

1-14. (canceled).

15. (currently amended) An apparatus, comprising:

a processor;

5 a media player;

a program code module comprising code executable by the processor to carry out the following method steps:

issuing a request to each of a set of servers and receiving a response to the request, and
using data associated with the response from each of the set of servers to identify a given server
10 in the set of servers;

as a media stream is being received from a first server and rendered by the media player, determining whether the media stream is acceptable according to a given metric;

if the media stream is not acceptable, and as the media stream continues to be received, taking a given action to initiate delivery of the media stream from ~~a second~~ the given server,
15 wherein the given action includes the steps of: (a) creating a buffer; (b) receiving from the first server and caching in the buffer advanced portions of the media stream; (c) issuing a request to the ~~second given~~ given server to initiate delivery of the media stream at a given offset; and (d) rendering the advanced portions of the media stream;

receiving the media stream from the ~~second given~~ given server; and

20 when the given offset is reached, rendering in the media player the media stream received from the ~~second given~~ given server.

16. (previously presented) The apparatus as described in claim 15, wherein the code is executable by the processor to initiate an instruction to the first server to cease transmission of
25 the media stream before rendering the advanced portions of the media stream.

17. (cancelled).

18. (currently amended) The apparatus as described in claim 15, wherein the code is
30 executable by the processor to match data packets received from the first ~~and second servers~~
server and the given server such that the media stream rendered in the media player appears

continuous.

19. (cancelled).

5 20. (previously presented) The apparatus as described in claim 15 wherein the media stream is not acceptable if it is being thinned by the first server.

21. (previously presented) The apparatus as described in claim 15 wherein the media stream is not acceptable if a given indication from the first server is received.

10

22. (previously presented) The apparatus as described in claim 21 wherein the given indication is that the first server will be unavailable.

15 23. (currently amended) The apparatus as described in claim 15, wherein the code executable by the processor determines that the ~~second~~ given server has a response time that differs from a response time of ~~the first server~~ at least one other server in the set of servers.

20 24. (previously presented) The apparatus as described in claim 15, wherein the code executable by the processor determines whether the media stream is acceptable periodically as the media stream is being delivered.

25 25. (previously presented) The apparatus as described in claim 15, wherein the code executable by the processor records given data associated with receipt of the media stream.

26. (previously presented) The apparatus as described in claim 15, wherein the advanced portions of the media stream are created by instructing the first server to increase a delivery rate of the media stream or by instructing the media player to decrease a rendering rate of the media stream.

30 27. (new) The apparatus as described in claim 15 wherein the request is an RTSP OPTIONS command.

28. (new) The apparatus as described in claim 15 wherein the request is associated with a DNS SRV protocol.

5 29. (new) The apparatus as described in claim 15 wherein the set of servers is identified by a content delivery network (CDN) map generating process.

30. (new) The apparatus as described in claim 29 wherein the method carried out by the program code constructs a header that identifies, for each of the set of servers, an IP address
10 for the server, together with a value indicating a time for the response to be returned after issuing the request.

31. (new) The apparatus as described in claim 31 wherein the method carried out by the program code transmits the header to the CDN map generating process to facilitate building
15 of a map of Internet traffic conditions.